

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-11 (Cancelled).

12. (Currently Amended) A method of creating multimedia services in a network comprising the steps of:

a) assembling interlocking graphical language blocks into a service logic script, said graphical language blocks having varying sizes of notches and varying number of extending bulges for enforcing rules for their connectivity, wherein said graphical language blocks each represent service control or call control functions and wherein each block has at least one input or output for passing a-an executable token between blocks, said token causing execution of said blocks;

b) installing said service logic script in a service execution environment;
and

c) translating said graphical language blocks into programming language objects when said service logic script is installed and executed.

13. (Original) The method according to claim 12, wherein a block begins execution when the block receives said token.

14. (Original) The method according to claim 13, wherein a plurality of blocks may execute simultaneously.

15. (Previously Presented) The method according to claim 14, wherein said graphical language blocks represent complex event driven actions which take place in the network and which are hidden from a user.

Claims 16-23 (Cancelled)

24. (Currently Amended): A service creation system for creating multimedia services, comprising:

- a service creation environment for creating a service logic script, the service creation environment further comprising:

- an editor for providing graphical capability to create a service logic script for a desired service using interlocking language graphical objects~~blocks~~ having varying sizes of notches and varying number of extending bulges for enforcing rules for their connectivity, wherein said language graphical ~~objects~~ blocks each represent service control or call control functions and wherein each block has at least one input or output for passing ~~a~~ an executable token between blocks, said token causing execution of said blocks;

- a translator for translating language graphical objects into lower level language executable objects; and

- a data repository for storing service and customer related data necessary for the desired system;

- a service execution environment for executing said service logic script;

and

- application programming interfaces between said service creation environment and said service execution environment.

25. (Original) The service creation system according to claim 24, wherein a block begins execution when the block receives said token.

26. (Original) The service creation system according to claim 25, wherein a plurality of blocks may execute simultaneously.

27. (Previously Presented) The service creation system according to claim 26, wherein the desired service is a call follow-me service.

28. (Previously Presented) The service creation system according to claim 26, wherein the desired service is a restricted calling service.

29. (Previously Presented) The service creation system according to claim 26, wherein said language graphical objects represent complex even driven actions which take place in a network and which are hidden from a user.

30. (Currently Amended) The service creation system according to claim 29, wherein said service ~~execution~~execution environment further comprises a service location, a service instantiator, and a service logic executor.

31. (Previously Presented) The service creation system according to claim 30, wherein said service locator identifies service logic based on a service subscriber identification.

32. (Previously Presented) The service creation system according to claim 31, wherein said service locator selects a service ID and finds a service logic execution environment in which the logic is stored.

33. (Previously Presented) The service creation system according to claim 30, wherein said service instantiator instantiates invoked service logic with required service and subscriber data so as to be ready to be executed in the service execution environment.

34. (Previously Presented) The service creation system according to claim 30, wherein said service logic executor loads executable service logic code, reads and stores service and subscriber data, provides inter-process communication between various service logic scripts that are being executed concurrently.

35. (Previously Presented) The method according to claim 15, wherein said service logic script captures all interactions with a service subscriber, requests network resources on behalf of the service subscriber, provides access to all required data, and prepares information for service billing purposes.

36. (Previously Presented) The method according to claim 15, wherein service control and call control functions of the graphical language blocks are mapped into service control and call control functions of the application programming interfaces (APIs), which are exposed in the service execution environment.

37. (Previously Presented) The method according to claim 15, wherein dependencies are established between service control function or call control functions of the graphical language blocks and event notifications sent from the service execution environment.

38. (Previously Presented) The method according to claim 15, wherein next state of the service control or call control function is determined upon its completion or upon receiving an event notification from the service execution environment.

39. (Previously Presented) The method according to claim 36, wherein the application programming interfaces provide access to and control of network resources which are used during service invocation by service subscribers.

40. (Previously Presented) The method according to claim 15, further comprising the step of:

determining if a service requires data and automatically storing, populating, and retrieving data during service instantiation.

41. (Previously Presented) The method according to claim 40, wherein said data comprises at least one of system data, service data, subscriber data, and local data.

42. (Previously Presented) The method according to claim 40, further comprising the step of:

generating service provisioning forms for the entry of required data.

43. (Previously Presented) The method according to claim 40, further comprising the step of:

generating subscriber tuning forms to allow a service subscriber to enter data related to the service subscriber.